iMOS: Enabling VoIP QoS Monitoring at Intermediate Nodes in an OpenFlow SDN

Christina Thorpe[‡], Adriana Hava[§], Julien Langlois[‡¶], Alexis Dumas[‡¶], Cristian Olariu[‡]
[‡]University College Dublin, School of Computer Science, Belfield, Dublin 4, Ireland
[§]Dublin City University, School of Electronic Engineering, Glasnevin, Dublin 9, Ireland
[¶]Ecole Centrale de Nantes, 1 rue de la No, BP 92101, 44321 NANTES cedex 3, France
christina.thorpe@ucd.ie, adriana.havaolariu@dcu.ie, julien.langlois@eleves.ec-nantes.fr
alexis.dumas@eleves.ec-nantes.fr, cristian.olariu@ucd.ie

***Abstract*—The growing popularity of outsourced enterprise VoIP services poses a significant quality assurance issue for service providers. VoIP traffic is very sensitive to network impairments and maintaining high QoS across multiple domains can be challenging. We propose to use SDN and our implementation of intermediate VoIP call quality measurement to provide an advanced VoIP monitoring service. Our solution can automatically detect and locate quality issues for VoIP traffic.**

Keywords*—*VoIP, SDN, OpenFlow, iMOS, QoS

I. Introduction

Hosted Enterprise VoIP (Voice over IP) services are becoming an extremely popular choice for small to medium sized enterprises. In this model, the phone service is outsourced to a third party service provider, and is typically delivered over the public Internet. Enterprise VoIP services provide organisations with advanced features and functionality such as Private Branch Exchange (PBX) capabilities, and conference calling.

It is well known that VoIP is extremely sensitive to network conditions [1], and thus, is one of the most difficult traffic types (in addition to video, remote surgery, control room traffic) for which to ensure a high Quality of Service (QoS). The quality of a VoIP call can degrade if there is loss, delay, or jitter on the network links between two end users. Traditional methods of VoIP QoS assessment include the Mean Opinion Score (MOS) - a metric used to grade the end-to-end quality of a call. However, simply knowing that the call quality is bad is not sufficient to enable dynamic improvement to the QoS. More granular monitoring is needed to locate the root cause of the quality degradation.

Intermediate MOS (iMOS) [2] is a VoIP QoS metric that can be used to calculate the MOS at intermediate nodes on the path of a VoIP flow. Locating the source of QoS degradation can inform corrective action that can improve the VoIP service. However, traditional network architectures are not flexible enough to support dynamic reconfiguration. Software Defined Networking (SDN) is a networking paradigm that gives administrators and service providers much more flexibility in terms of determining the behaviour of the network and the transport quality of the services that run over it. Native OF stats are insufficient for an accurate VoIP user perception evaluation and the iMOS metric implemented in this work was created to address this issue.

OpenFlow (OF) [3] is one embodiment of SDN that has reached significant penetration and is supported by many hardware vendors. There are many OF controllers that can be programmed with intelligence to enforce decisions based on defined business logic. For example, Call Admission Control (CAC), rerouting, and quality adaptation can all be enforced on VoIP flows in SDN. Reconfiguration can be triggered automatically by the controller based on the QoS of the VoIP traffic traversing the network.

This paper proposes an approach for advanced VoIP monitoring in SDN. The iMOS quality indicator was implemented using OF to provide an insight into the per-hop VoIP quality. Modifications were made to the OF switches and an application was written for the Floodlight controller to support the iMOS. Comprehensive experimentation was undertaken to validate the new functionality. This work builds on our previous experience in the area of intermediate VoIP monitoring [2], advanced QoE metrics for IPTV [4], and assuring high Quality of Experience (QoE) in an IP Television (IPTV) network using an OpenFlow SDN [5], [6].

The remainder of the paper is organised as follows: Section II discusses a selection of the related work in the literature. Section III presents an overview of the pertinent technical details of the technologies employed in this work. Section IV details the iMOS implementation in OF to enable advanced VoIP monitoring. Section V describes the modifications made to both the OF switches and the Floodlight controller during the construction of the testbed. Section VI presents a series of experiments and results used to validate the new functionality implemented in OF. Finally, section VII concludes this paper.

II. Related Work

There has been a considerable amount of activity in the SDN research community in recent years. Phemius et. al. [7] propose a mechanism to measure link latencies from an OpenFlow controller with high accuracy and a low footprint to overcome the challenges involved with latency determination in OpenFlow. Van Adrichem et. al. [8] present an approach and open-source software implementation to monitor per-flow metrics, such as throughput, delay and packet loss, in OpenFlow networks. Qazi et. al. [9] proposed Atlas, which incorporates application-awareness into SDN. Kumar et. al. [10] looks at improving the QoE of services in home networks using SDN. Kwon et. al. [11] propose the adaptive Mobile

Voice over Internet Protocol (mVoIP) service architecture in SDN networks to provide the best quality of mVoIP service to end-users.

Our work is different from the existing literature as it proposes the implementation of a VoIP monitoring solution, called iMOS, to measure the quality of the VoIP calls at intermediate points in the SDN. The most difficult part of monitoring VoIP calls is obtaining the network delay. The solution in [8] uses an intrusive solution to measure the delay, whereas our solution uses information obtained from the actual VoIP call to estimate the network delay.

III. Technical Background

A. OpenFlow

SDN is a relatively recent networking paradigm that facilitates new innovative and flexible networking environments. SDNs provide an opportunity for network administrators and service providers to add new service-specific functionality for monitoring and management in order to improve the QoS/QoE for their subscribers [12]. Currently, the most common protocol for communication between the controller and the data plane is OpenFlow, and many network vendors already implement it in their components.

OpenFlow is an open standard, that allows researchers and developers to add network functionality and run experiments, without needing to know the internal workings of hardware network devices. OF Switches (OFSs) are composed of one or more flow tables. The OF Controller (OFC) imposes policies on the switch flows. Flows contain a set of packet fields that are used to match flows, and a set of actions (e.g. send-out-port, modify-field or drop) used to process the packet. The OF protocol consists of several messages used to define the behaviour of the network. An OFS has several flow tables; when a packet is received, it enters the OF processing pipeline as described in the following paragraph.

If a packet arriving at the switch matches an entry in the first flow table, a look-up is performed for the corresponding instruction: (i) The packet can be modified with an 'apply-action' instruction. (ii) The action set (initially empty) can be updated with a 'write-action' or 'clear-action' instruction. (iii) The metadata is updated with a 'write-metadata' instruction. (iv) The packet can be sent to a following table with a 'goto' instruction. (v) If there is no 'goto' instruction, the action set is executed and the packet leaves the switch. (vi) If the packet doesn't match any entry in the table, it is sent to the controller or dropped.

B. Enterprise VoIP

Figure 1 illustrates a typical enterprise VoIP service architecture. It consists of a Cloud PBX hosted by the enterprise VoIP service provider. Small and Medium Enterprises (SMEs) subscribing to this service are connected to the PBX via business-grade broadband links to the Internet. The office infrastructure can vary and it depends on the number of employees. Larger offices such as the main office typically will have many IP phones connected to a data switch. Traffic will flow through the switch, then the VoIP gateway, and finally leave the premises via the Asymmetric Digital Subscriber Line

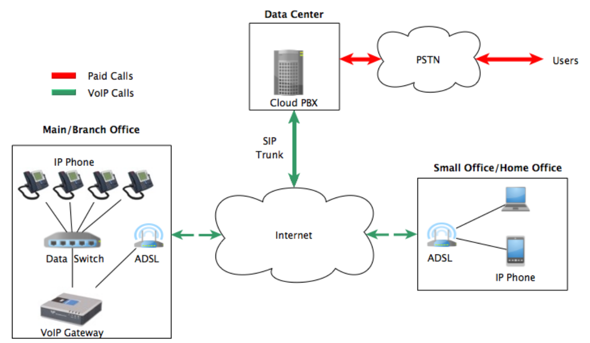

Fig. 1. Typical Enterprise VoIP Architecture

(ADSL) gateway. The Cloud PBX is connected to the Internet via Session Initiation Protocol (SIP) trunks. Subscribers benefit from free VoIP to VoIP calls, whereas typically VoIP to Public Switched Telephone Network (PSTN) calls incur costs.

C. VoIP Quality

Although VoIP quality can be affected by codecs, hardware, and bandwidth, the quality of a call can be defined by 3 main network measures:

- Data loss: the percentage of packets lost during a measurement period.
- Delay or latency: The length of time taken for a packet to traverse the network from source to destination. Delay can be broken down into 4 parts: (i) Processing Delay: the time it takes for the router to process the packet. (ii) Transmission Delay: the time it takes to push all the bits of the packet onto the link. (iii) Propagation Delay: the time it takes for the packet to reach the next hop on the path. (iv) Queueing Delay: the time the packet spends in routing queues. This paper defines delay to equate to the sum of these delays.
- Jitter [13]: This represents the variation in the delay between adjacent packets. High jitter indicates that there is congestion on the network.

1) MOS: Traditionally, the quality of a VoIP call is graded using a MOS score, which is based on the human user's perception of the quality of the network. The MOS is in the range of 1 to 5 where 5 represents the best quality (see Table I).

TABLE I. MOS Score

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

However, it is not feasible or practical to monitor VoIP call quality using human subjects. Hence, network QoS metrics are used to estimate the call quality. The most popular model for these estimations is the E-Model [14], which proposes an algorithm to compute end-to-end voice quality rating. The primary output is a scalar called the Transmission Rating Factor. The standard E-Model is not suitable for real-time measurements, but [15] proposes an adaptation of the model that can be used to compute a MOS score in real-time.

2) iMOS: End-to-end MOS is relatively simple to compute using information that is available on each terminal. However, for real-time quality assessment of VoIP calls, the MOS must be evaluated at intermediate nodes in the network. For example, in an SDN, the OFC can use information about the call (including the MOS) to improve the data path. The MOS computed at an intermediate network component is called the intermediate Mean Opinion Score (iMOS). The principle is the same as the traditional MOS: delay, jitter and loss are important factors, but these are more complex to calculate at intermediate points.

IV. iMOS: VoIP MOS Measurements at Intermediate Network Nodes

The MOS for VoIP conversations can be obtained using the formula proposed by the E-Model [16]. The output of the formula is the transmission factor *R*, which is obtained using the following equation: $R = R_0 - I_s - I_d - I_{e\_eff} + A$, where R_0 is the signal to noise ratio; I_S is the impairment that occurs more or less simultaneously with the voice signal (e.g., speech volume too loud); I_d is the impairment due to a delay or echo effect; I_{e_eff} is the impairment due to low bit-rate codecs and packet-losses; and A is the advantage factor that allows factoring in the user's wiliness to accept lower call quality for ease of service access (e.g. calls from remote locations). In a study done after the proposal of the E-Model, Clark [17] has shown that, for IP-based calls, most of the telephone line specific terms can be reduced to their default values, thus simplifying the E-Model to encompass only IP network specific parameters such as packet delay, loss, and jitter. The resulting equation is: $R = 94 - I_d - I_{e\_eff} + A$.

The E-Model was initially designed to be used by end-nodes, where the VoIP packets originate or terminate. All metrics needed by the model are easily obtained at end-nodes, however this is different at intermediate nodes. More specifically, the delay is much harder to measure or to estimate at intermediate points, as the classical estimation using the RTT of the RTCP packets becomes unusable in this situation.

We present a solution to this critical issue that enables intermediate points to estimate the MOS of VoIP conversations. Our solution is based on the RTCP reports sent periodically by the call participants. The report contains useful information that conversation parties use to obtain an estimation of their peers' perceived quality and is also used to keep the session clock in sync.

Two fields in particular are useful to our proposal: the RTP [13] timestamp field that carries information about the current session clock of the VoIP call, and the NTP timestamp field that carries information about the local 'wall clock' time that corresponds to the session clock. The VoIP session clock is a counter increased with every sample taken by the codec. Typically codecs sample the audio signal at a fixed rate (e.g. 8 kHz for G711), and the timestamp included in the RTP header reflects the sampling instant of the first byte in that RTP packet.

Intercepting at least one RTCP [13] packet, will enable an intermediate node to estimate the delay that affected any RTP packet from its generation until it was intercepted in the intermediate node. This process is depicted in Figure 2, and the delay of any RTP VoIP packet can be obtained using (1).

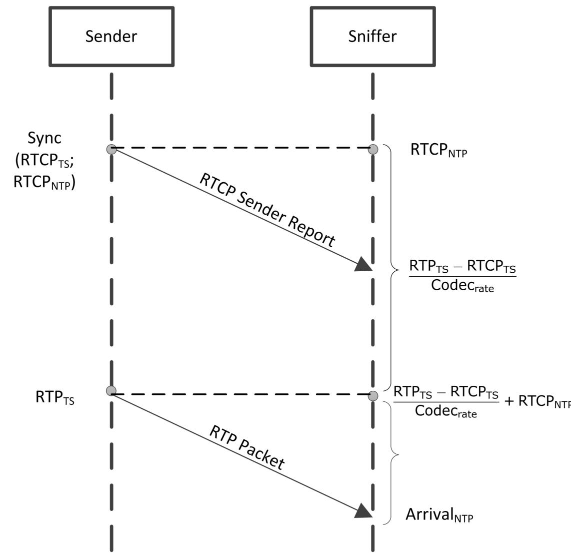

Fig. 2. Using RTCP NTP timestamps to compute the delay.

Fig. 2 contains a sequence diagram that illustrates the delay computation based on RTCP NTP timestamps. $RTCP_{TS}$ is the session timestamp when the RTCP packet originated; $RTCP_{NTP}$ is the wall clock time when the RTCP packet originated; RTP_{TS} is the session clock timestamp found in RTP packets following the RTCP packet; $Arrival_{NTP}$ is the wall clock time when the RTP packet was intercepted; and finally, $Codec_{rate}$ is the VoIP codec sampling rate.

$$\text{delay} = \text{Arrival}_{\text{NTP}} - \left(\frac{\text{RTP}_{\text{TS}} - \text{RTCP}_{\text{TS}}}{\text{Codec}_{\text{rate}}} + \text{RTCP}_{\text{NTP}} \right) \tag{1}$$

Equation 1 is based on two assumptions: i) end-nodes are using NTP or PTP to set and adjust the wall clock of their communication device, and ii) the codec sampling rate is constant. While the former is true in most cases, especially in the controlled hosted VoIP services, the latter does not hold for bitrate voice codecs. The uncertainty of knowing the codec sampling rate in the current measurement period leads to situations where jitter is introduced and the estimated delay can have negative values. To alleviate this problem, we adjust the apparent codec rate with it's variation from the codec's nominal rate. Further, our non-intrusive solution is based on existing VoIP packets traversing the network, while there are other intrusive solutions, e.g. [18] inject measurement probes to estimate link delay, however intrusive methods are widely criticised for the traffic overhead created.

V. Software Defined Networking Testbed

This section discusses the modifications made to both OF and the Floodlight controller to support our VoIP monitoring solution: (i) New modules were added to the OF switch to support the iMOS metric (ii) Modifications were made to the Floodlight controller to support VoIP monitoring on the switches and to display the VoIP iMOS on the GUI.

A. OpenFlow iMOS-Enabled Switch

In order to determine VoIP QoS at intermediate nodes in a SDN network, each OF switch must be able to compute the iMOS metric for any VoIP call. Thus, the OF nodes have been upgraded with adjustments to the existing OF Report Collection Module and the inclusion of our iMOS Metric module. The architecture of our proposed OpenFlow node is depicted in Figure 3.

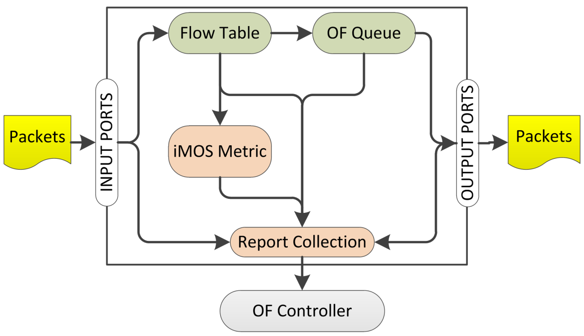

Fig. 3. OF node with iMOS metric module and Report Collection module

The Flow Table module identifies the VoIP packets, using rules set by the controller, and forwards a copy of the VoIP packets to the VoIP iMOS module. Two different types of packets are tracked by the VoIP iMOS module: SIP packets, which are used to establish a VoIP session, and the actual VoIP packets with are carried over RTP.

The VoIP iMOS module sends the computed iMOS value to the Report Collection module, from where it can be requested by the SDN controller. The VoIP iMOS module can be enabled or disabled by the SDN controller. This means that, in an SDN network, iMOS-enabled switches can inter-operate with non-iMOS-enabled switches. The newly added module does not interfere with the default operation of an OpenFlow switch.

For each call, SIP information and RTP stream information is extracted. SIP packets provide information about the VoIP sessions, such as: call-ID, codec, rate, and call start time.

Table II details the the structure of the packet storing SIP information:

TABLE II. SIP INFORMATION

Call-ID	IP Caller	IP Receiver	Codec Type	Codec Rate	Call Request	Call Status	Call Time
4B	4B	4B	2B	4B	1B	2B	4B

RTP packets identify the VoIP streams running through the OF switch. For each VoIP stream identified using the SSRC field in the RTP header, the packet loss, jitter, and delay are use to compute the iMOS.

B. Floodlight Controller Upgrade

As mentioned in section III, the OF controller is the SDN component that imposes different forwarding policies on the OF switches. In our test-bed, the FloodLight controller is used as it provides an intuitive web GUI for visualising the network. By default, the Floodlight controller is not VoIP aware, thus new functionality was implemented in it.

On the GUI side, each switch displays additional information related to the VoIP calls passing through it. To retrieve this information from the switches, a set of new classes were added to Floodlight. These classes extend the Statistic class and extract the information available in the Report Collection module of the switches. Thus, through the Floodlight's GUI, information about a VoIP call (e.g., source and destination IP, call ID, RTP port, delay, jitter, loss, and iMOS) is displayed.

Another upgrade to the FloodLight controller is the ability to enable and disable the iMOS metric computation functionality on the switches. A new OF message was defined, which is sent by the Floodlight controller when the metric needs to be activated or deactivated. For the functionality mentioned above, the messages to query or send instructions to the switches were created using the OF Vendor Message format that allows the communication of custom OF messages between the controller and the switches.

C. Test-Bed Setup

The SDN test-bed deployed for testing the accuracy of the iMOS metric is depicted in Figure 4 and comprises of the following components:

- VoIP Clients (h0, h1, h2, and h3) running on separate laptops, that use Linphone - an open-source VoIP software for voice conversations that uses SIP and Opus codec.
- OF Switches (s0, s1, s2, and s3) that run the iMOS module. OF switches receive custom OF messages from the controller to enable/disable the iMOS metric.
- SDN Floodlight Controller (c0) which was improved to support the iMOS related communication with the switches. Additionally, the controller can request iMOS statistics from the OF switches and display them through its GUI.

The tests were run on a machine equipped with an Intel i7-3610QM@2.30Ghz processor, 6GB of RAM, running Ubuntu 14.04 64bits, and were repeated 5 times to ensure repeatability and stability of the observed metrics.

VI. VALIDATION

A. Impact of Codec Sampling Rate on Delay Accuracy

The delay estimation detailed in section IV is based on the assumption that the codec sampling rate is constant. However, this is not the case for variable bitrate codecs; thus, we proposed an apparent codec rate adjustment. Figures 5 and 6 show the accuracy of the delay measurement when a variable bitrate VoIP call is transported on a communication link that delays all packets by 50 millisecond. When rate adjustment is not used, the jitter over-imposed on the delay measurements is high, as can be seen in Figure 5. The jitter diminishes (Figure 6) when rate adjustment is used, however some artifacts are still present, but with limited impact on the iMOS.

B. Negative Delay Effect

This experiment shows that negative delay values can be obtained when the codec rate is under-adjusted. Figure 7 shows

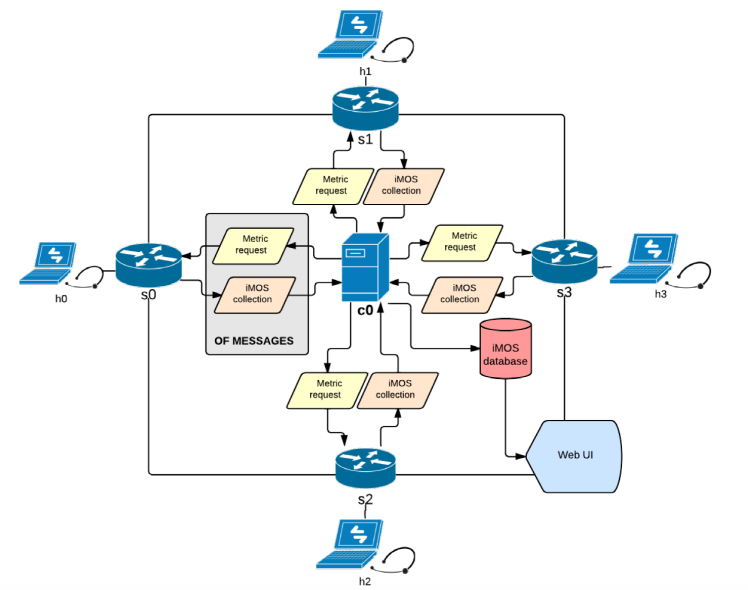

Fig. 4. SDN Test-Bed

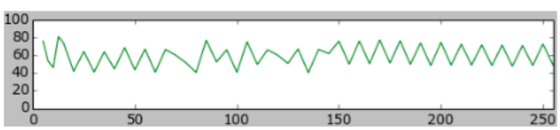

Fig. 5. Delay estimation without codec rate adjustment when the actual delay is 50 milliseconds. X axis is time (s), and Y axis is estimated delay (ms) calculated with equation (1)

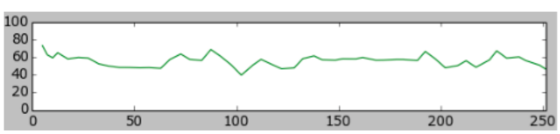

Fig. 6. Delay estimation with codec rate adjustment when the actual delay is 50 milliseconds. X axis is time (s), and Y axis is estimated delay (ms) calculated with equation (1)

that there are many negative samples when a delay of 10 milliseconds is affecting every VoIP packet. This also has a negative effect on the iMOS estimation. When codec rate adjustment is enabled (Figure 8) some negative values can still appear, however their impact on the iMOS estimation is minimal when compared to the previous case.

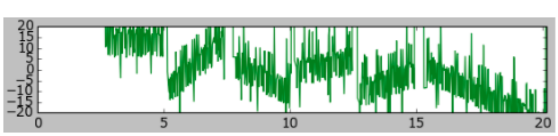

Fig. 7. Negative delay effect without codec rate adjustment when the actual delay is 10 milliseconds. X axis is time (s). Y axis is delay (ms)

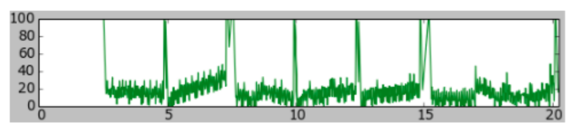

Fig. 8. Negative delay effect with codec rate estimation when the actual delay is 10 milliseconds. X axis is time (s). Y axis is delay (ms)

C. Actual Versus Estimated Delay

We investigated the accuracy of our delay estimation at intermediate points against the actual delay that affects VoIP packets in a range from 0 to 100 milliseconds. Figure 9 depicts the results; the green line represents the estimated delay. It can be seen that the difference between the actual delay and the estimated one decreases when the actual delay increases. This is a good result since the delay has significant impact on the iMOS when its value is high. The estimation error at small delay values can only lead to false positives, which is beneficial if the scope of using the iMOS is, for example, CAC.

D. iMOS Versus End-to-End MOS

Finally, we compared the end-to-end MOS values with iMOS values obtained at an intermediate OF switch when network conditions are dynamic. We varied the network delay, and Figure 10 shows that the iMOS is reported to be slightly higher than the end-to-end MOS. This is due to the fact that the iMOS captures the quality at an intermediate point and does not account for the factors degrading the quality of the call between that point and the receiver of the call. This result validates the usefulness of the proposed solution. Researchers and engineers can leverage our solution to observe how VoIP quality degrades along a communication line, and eventually pin-point the node or link that caused the highest quality drop. Figure 10 shows that between the intermediate points where the iMOS is collected and the receiver of the call, there has been a 0.5 MOS decrease.

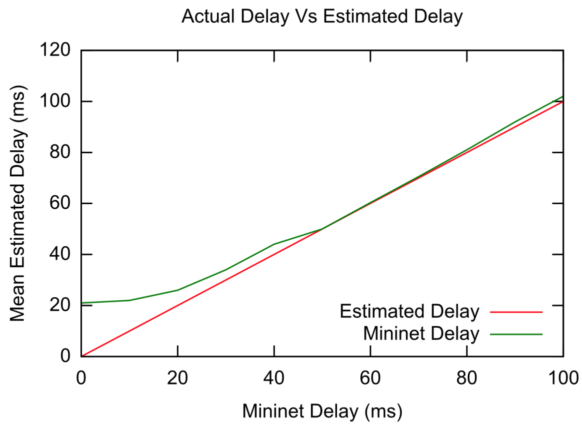

Fig. 9. Error of estimated versus actual delay

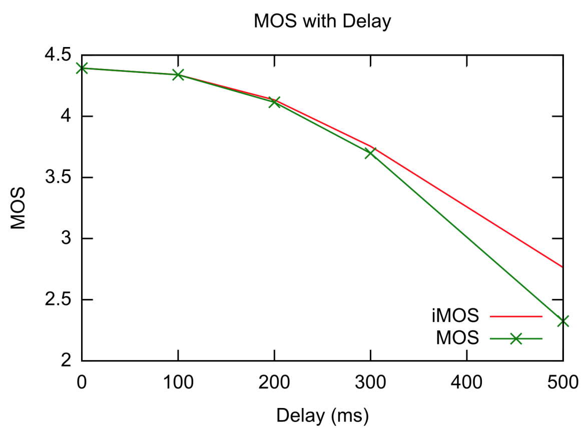

Fig. 10. Impact of delay on end-to-end MOS versus iMOS

VII. Conclusion

This paper details an approach to managing enterprise VoIP services using Software Defined Networking. Various different technologies, used to create a cost-effective prototype for research purposes, were discussed. The paper includes a comprehensive description of advanced VoIP call quality monitoring in SDN, implemented using OF and Floodlight. To date, several different novel corrective mechanisms have been developed and validated on the testbed, which are currently the subject of further investigation and development. The future work for the proposed VoIP aware SDN includes the implementation of network reconfiguration such as a CAC mechanism that can prevent congestion on a network node caused by too many active calls, and to investigate how our solution scales with higher volume of VoIP traffic, and what adjustments are needed to consider VoIP traffic over TCP rather than UDP.

Acknowledgment

This work was supported, in part, by Science Foundation Ireland grant 10CEI1855 and in part by Science Foundation Ireland grant 13RC2094.

References

[1] A. P. Markopoulou, F. Tobagi, M. J. Karam *et al.*, "Assessment of voip quality over internet backbones," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2002, pp. 150–159.

[2] C. Olariu, J. Fitzpatrick, P. Perry, and L. Murphy, "A QoS based call admission control and resource allocation mechanism for LTE femtocell deployment," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 884–888.

[3] "OpenFlow Specification Simulation Tool." [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

[4] R. de Fréin, C. Olariu, Y. Song, R. Brennan, P. McDonagh, A. Hava, C. Thorpe, J. Murphy, L. Murphy, and P. French, "Integration of qos metrics, rules and semantic uplift for advanced iptv monitoring," *Journal of Network and Systems Management*, pp. 1–36, 2014.

[5] P. McDonagh, C. Olariu, A. Hava, and C. Thorpe, "Enabling iptv service assurance using openflow," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, 2013, pp. 1456–1460.

[6] C. Thorpe, C. Olariu, A. Hava, and P. McDonagh, "Experience of developing an openflow sdn prototype for managing iptv networks," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 966–971.

[7] K. Phemius and M. Bouet, "Monitoring latency with openflow," in *Network and Service Management (CNSM), 2013 9th International Conference on*. IEEE, 2013.

[8] N. L. Van Adrichem, C. Doerr, F. Kuipers *et al.*, "Opennetmon: Network monitoring in openflow software-defined networks," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014.

[9] Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir, "Application-awareness in sdn," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 487–488.

[10] H. Kumar, H. H. Gharakheili, and V. Sivaraman, "User control of quality of experience in home networks using sdn," in *Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[11] D. Kwon, R. Thay, H. Kim, and H. Ju, "Qoe-based adaptive mvoip service architecture in sdn networks," *ICN 2014*, p. 73, 2014.

[12] C. Monsanto, J. Reich, N. Foster, J. Rexford, D. Walker *et al.*, "Composing Software Defined Networks," in *NSDI*, 2013, pp. 1–13.

[13] H. Schulzrinne, S. Casgner, R. Frederick, and V. Jacobson, "RFC 3550, RTP: A Transport Protocol for Real-Time Applications," 2003. [Online]. Available: https://tools.ietf.org/html/rfc3550

[14] J. A. Bergstra and C. A. Middelburg, "ITU-T Recommendation G.107 : The E-Model, a computational model for use in transmission planning," Tech. Rep., 2003.

[15] Psytechnics, "Estimating E-Model Id within a VoIP Network," Tech. Rep., 2002.

[16] ITU-T G.107, "The E-Model, A Computational Model For Use In Transmission Planning," 2005.

[17] A. D. Clark, "Modeling the effects of burst packet loss and recency on subjective voice quality," 2001.

[18] C. Yu, C. Lumezanu, A. Sharma, Q. Xu, G. Jiang, and H. V. Madhyastha, "Software-defined latency monitoring in data center networks," in *Passive and Active Measurement*. Springer, 2015, pp. 360–372.